

# 10

## DESAIN RANGKAIAN BERURUT

Desain rangkaian berurut pada umumnya dimulai dari uraian dengan kata-kata (verbal) tentang perilaku (behaviour) daripada rangkaian berurut yang akan dibangun. Uraian kebutuhan (spesifikasi) ini dituangkan ke dalam bentuk diagram keadaan atau tabel keadaan. Dalam banyak kasus, tabel keadaan yang diperoleh dalam langkah pertama ini belumlah menghasilkan cacah keadaan yang minimum sehingga masih membutuhkan penyederhanaan. Penyederhanaan keadaan dapat dilakukan dengan 2 metoda, metoda pencocokan baris (row matching) atau metoda peta pasangan (pair chart). Berdasarkan kebutuhan cacah keadaan minimum ini, maka cacah flip-flop yang dibutuhkan telah tertentu. Untuk kebutuhan sebanyak  $(2^{n-1} + 1)$  s/d  $2^n$  keadaan, dibutuhkan  $n$  buah flip-flop.

Persamaan masukan (eksitasi) masing-masing flip-flop dari jenis yang dipilih dapat diturunkan dari tabel keadaan. Dengan diketahuinya persamaan masukan, maka diagram rangkaian yang diinginkan telah dapat digambarkan. Tentunya rangkaian yang diperoleh masih perlu ditilik apakah sudah melaksanakan spesifikasi rancangan.

Dari uraian di atas dapat dilihat bahwa walaupun sarana yang digunakan sama, yaitu tabel keadaan atau diagram keadaan dan persamaan masukan, urutan langkah-langkah dalam desain merupakan kebalikan dari pada urutan langkah-langkah yang ditempuh dalam analisis yang telah diuraikan dalam bab sebelumnya. Dalam bab ini akan diuraikan langkah-langkah pembentukan tabel keadaan dan diagram keadaan berdasarkan spesifikasi kebutuhan diikuti penyederhanaannya dan dilanjutkan dengan penentuan persamaan masukan dan realisasinya.

### 10.1 Desain Pencacah

Untuk memahami langkah-langkah desain yang diterangkan di atas, kita coba terapkan metoda tersebut dalam merancang pencacah yang telah diuraikan dalam Bab 7.

Misalkan kita ingin membuat suatu pencacah untuk perekaman perolehan nilai (scorer). Bila jawaban benar, maka nilai perolehan ditambah sedangkan bila jawaban salah maka nilai perolehan dikurangi sebesar setengah dari penambahan nilai bagi jawaban benar. Untuk mudahnya, anggaplah nilai bagi jawaban benar adalah 2 dan nilai bagi jawaban salah adalah negatif 1. Nilai terbatas antara 0

dan 5. Penambahan nilai diluar batas ini hendaknya ditunjukkan sebagai adanya limpahan (overflow), jawab benar pada saat nilai sudah berharga 4 membuat nilai menjadi 0. Tetapi jawab benar pada saat nilai sedang berharga 5 membuat nilai menjadi 1. Jawaban salah pada saat nilai sedang 0 akan membuat nilai menjadi 5 dan keadaan ini juga dianggap melimpah. Keadaan  $Z=1$  ini tentunya dapat diolah lebih lanjut, misalnya menggerakkan flip-flop lain sebagai tanda/bendera (flag) bahwa telah dicapai nilai 6 atau nilai negatif. Tetapi pengolahan lanjutan ini tidak akan kita bahas di sini.

Dari spesifikasi ini dapat disimpulkan bahwa kita membutuhkan suatu pencacah yang nilai dikandungnya akan naik atau turun setiap munculnya penabuh, tergantung keadaan sinyal jawaban  $x$  sebagai masukan luar. Harga dikandung (cacahan) naik 2 bila  $x=1$  dan turun 1 bila  $x=0$ . Harga cacahan kita batasi antara 0 dan 5. Bila cacahan dipaksa naik dari cacahan 4, cacahan akan melimpah dan kembali berharga 0. Tetapi bila cacahan dipaksa naik dari cacahan 5, cacahan akan melimpah dan berubah menjadi 1. Bila cacahan dipaksa turun dari 0, cacahan akan berubah menjadi 5, juga dianggap melimpah. Keadaan melimpah kita deteksi dengan membuat keluaran  $Z$  berkeadaan 1.

Untuk membangun diagram keadaan, kita misalkan keadaan awal pencacah adalah keadaan sebelum ada masukan, dengan cacahan berharga 0 dan kita sebut saja keadaan 0. Keadaan lain juga kita namakan sesuai dengan harga cacahannya.

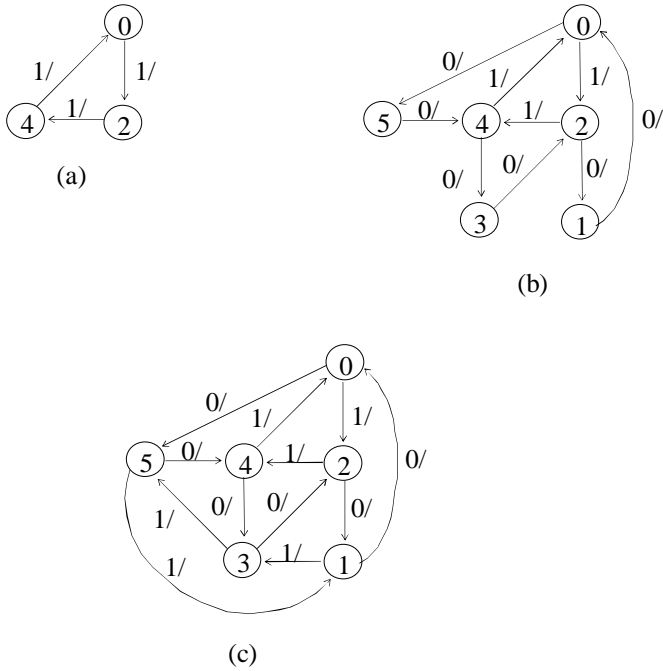
Kita mulai membuat diagram keadaan dengan menganggap secara berturut-turut terjadi masukan  $x=1$  hingga terjadi limpahan, yaitu keadaan berubah dari 0 ke 2, ke 4 dan kembali 0. Pada saat keadaan berubah kembali ke 0, keluaran  $z=1$ . Deretan perubahan ini digambarkan pada Gambar 10.1 (a).

Sekarang kita tinjau keadaannya bila pada keadaan 0 secara berturut-turut terjadi masukan  $x=0$  hingga keadaan kembali ke 0, yaitu keadaan berubah dari 0 ke 5, ke 4, ke 3, ke 2, ke 1, dan kembali ke 0. Keadaan melimpah terjadi saat keadaan berubah dari 0 ke 5. Dengan menambahkan perubahan ini kepada Gambar 10.1 (a), maka kita akan memperoleh diagram Gambar 10.1(b). Perhatikan bahwa dengan  $x=0$  terjadi secara berturut-turut, pencacah kita berfungsi sebagai pencacah turun modulus 5.

Pada Gambar 10.1 (b) semua keadaan telah lengkap dari 0 sampai dengan 5. Langkah terakhir adalah menilik apakah perubahan keadaan untuk semua kemungkinan harga  $x$ , yaitu  $x=0$  dan  $x=1$ , telah tergambar untuk setiap simpul, dan kita lengkapi yang belum lengkap. Dalam hal ini, yang perlu kita tambahkan adalah perubahan dari 1 ke 3, dari 3 ke 5, dari 5 ke 1, untuk  $x=1$ . Setelah menambahkan ini, maka kita akan memperoleh diagram akhir seperti pada Gambar 10.1(c).

Perubahan keadaan yang digambarkan dalam diagram keadaan pada

Gambar 10.1 (c) dapat kita tuangkan dalam tabel keadaan yang ditunjukkan sebagai pada Gambar 10.2 (a) atau Gambar 10.2 (b). Dengan Tabel Keadaan ini sudah dapat dicari persamaan masukan flip-flop dan persamaan keluaran.



Gambar 10.1. Diagram Keadaan Pencacah naik-turun

Keadaan sekarang	Keadaan berikut		Keluaran sekarang	
	x=0	x=1	x=0	x=1
0	5	2	1	0
1	0	3	0	0
2	1	4	0	0
3	2	5	0	0
4	3	0	0	1
5	4	1	0	1

ABC	A <sup>+</sup> B <sup>+</sup>		Z	
	x=0	x=1	x=0	x=1
000	101	010	1	0
001	000	011	0	0
010	001	100	0	0
011	010	101	0	0
100	011	000	0	1
101	100	001	0	1

(a)

(b)

Gambar 10.2. Tabel Keadaan Pencacah naik-turun Gambar 10.1.

Dengan menggunakan flip-flop T, maka tabel kebenaran masukan flip-flop dan keluaran dapat disusun seperti pada Gambar 10.3 (a) dan (b). Persamaan masukan dan keluaran untuk pencacah ini diperoleh dari Gambar 10.3 (c) sebagai berikut:

$$T_A = \bar{x}\bar{B}\bar{C} + xA + xB$$

$$T_B = \bar{x}A\bar{C} + x\bar{A} + \bar{B}C$$

$$T_C = \bar{x} \quad \text{dan} \quad Z = xA + \bar{x}\bar{A}\bar{B}\bar{C}$$

Rangkaian logika realisasi persamaan ini dapat disusun dengan mudah.

A B C	A <sup>+</sup> B <sup>+</sup> C <sup>+</sup>		T <sub>A</sub>		T <sub>B</sub>		T <sub>C</sub>	
	x=0	x=1	x=0	x=1	x=0	x=1	x=0	x=1
0 0 0	101		1	0	0	1	1	0
0 0 1	000		0	0	0	1	1	0
0 1 0	001		0	1	1	1	1	0
0 1 1	010		0	1	0	1	1	0
1 0 0	011		1	1	1	0	1	0
1 0 1	100		0	1	0	1	1	0
1 1 0	xxx	xxx	x	x	x	x	x	x
1 1 1	xxx	xxx	x	x	x	x	x	x

(a)

xA \ BC	00	01	11	10
00	1	0	1	0
01	0	0	1	0
11	0	x	x	0
10	0	x	x	0

$$Z = xA + \bar{x}\bar{A}\bar{B}\bar{C}$$

(b)

xB \ BC	00	01	11	10
00	1	1	1	0
01	0	0	1	0
11	0	x	x	1
10	0	x	x	1

T<sub>A</sub>

$$T_A = \bar{x}\bar{B}\bar{C} + xA + xB$$

xA \ BC	00	01	11	10
00	0	1	0	1
01	0	0	0	1
11	0	x	x	1
10	1	x	x	1

T<sub>B</sub>

$$T_B = \bar{x}A\bar{C} + x\bar{A} + \bar{B}C$$

xA \ BC	00	01	11	10
00	1	1	0	0
01	1	1	0	0
11	1	x	x	0
10	1	x	x	0

T<sub>C</sub>

$$T_C = \bar{x}$$

(c)

Gambar 10.3. Persamaan masukan dan keluaran pencacah pada

Gambar 10.1 dengan flip-flop T

## 10.2 Desain Detektor Urutan

Dalam sistem digital sering dibutuhkan rangkaian logika yang keadaannya tergantung atas urutan masukan yang telah terjadi, bukan keadaan masukan yang terakhir saja. Rangkaian demikian disebut detektor urutan (sequence detector) yang memberikan keluaran  $Z=1$  (atau  $Z=0$ ) hanya bila telah menerima deretan masukan yang telah mengikuti pola tertentu, sedangkan untuk pola lain keluaran tetap  $Z=0$  (atau  $Z=1$ ). Pencacah yang diuraikan dalam subbab sebelumnya tidak disebut detektor urutan karena  $Z=1$  terjadi bukan diakibatkan telah terjadinya suatu pola masukan tetapi karena terjadinya masukan tertentu pada keadaan tertentu. Rangkaian demikian banyak dijumpai dalam rangkaian kendali dalam industri, misalnya urutan pengaktifan rele (relay) untuk menutup kontaktor-kontaktor. Contoh lain adalah sistem kunci elektronik yang membuka kuncinya hanya setelah diaktifkan sederetan kombinasi saklar dalam urutan yang telah diprogram sebelumnya.

Misalkan kita hendak mendeteksi terjadinya deretan 010 pada saluran masukan seri suatu register geser. Kita inginkan agar setiap terjadinya pola tersebut ditunjukkan dengan keluaran yang berkeadaan 1 dan pola-pola masukan lain membuat keluaran berkeadaan 0.

Dari spesifikasi soal yang dihadapi dapat kita ketahui bahwa rangkaian akan memberikan keluaran 1 hanya bila masukan 0 diterima menyusul dua masukan berurutan 0 dan 1 sebelumnya. Perhatikan bahwa 0 terakhir dalam suatu pola deretan boleh jadi merupakan 0 pertama dalam pola deretan berikutnya. Perhatikan juga bahwa menerima dua atau lebih masukan 0 secara berturut-turut sama saja dengan baru menerima satu 0 dan menerima dua atau lebih 1 secara berturut-turut sama saja dengan belum menerima masukan sama sekali, karena pola dicari mulai dengan hanya satu 0.

Untuk merancang rangkaian yang dimaksud, kita perlu membuat sederetan masukan contoh yang mengandung pola yang akan dideteksi di antara pola-pola lainnya, dengan mulai dari keadaan belum menerima masukan sama sekali (keadaan reset). Hal ini perlu untuk meyakinkan bahwa kita telah mengerti sepenuhnya apa yang harus dilakukan oleh rangkaian itu. Untuk soal yang dihadapi ini, kita dapat menggunakan contoh deretan masukan berikut ini, yang disertai keadaan keluaran yang bersesuaian.

Input X : 0 1 1 0 1 0 1 0 0 1 1 1 0 1 0 1 1 0 0  
 Output Z : 0 0 0 0 0 1 0 1 0 0 0 0 0 0 1 0 0 0 0

Berbeda dengan soal pencacah yang diuraikan sebelumnya dimana cacah keadaan telah tertentu, dalam soal ini kita tidak tahu sebelumnya berapa keadaan

dalam rangkaian dan berapa sebenarnya flip-flop yang diperlukan untuk merealisasikan. Cacah itu baru akan diketahui setelah desain selesai. Untuk itu secara sembarangan kita pilih suatu keadaan tertentu sebagai keadaan awal yang kita namakan  $S_0$  dan untuk setiap kemungkinan masukan berikutnya kita tandai sebagai keadaan  $S_1$ ,  $S_2$ , dan seterusnya. Dari spesifikasi soal yang dihadapi dapat diketahui bahwa di samping keadaan belum menerima masukan, kita membutuhkan setidaknya 3 keadaan yang menandai telah terjadinya masukan pembentuk pola yang akan dideteksi, yaitu

- telah ada satu masukan 0,  $Z=0$
- telah terjadi urutan masukan 01,  $Z=0$  dan
- telah terjadi urutan 010 dan keluaran harus  $Z=1$

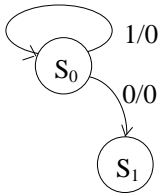
Kita andaikan saja keadaan awal  $S_0$  sebagai keadaan sebelum menerima masukan. Bila dalam keadaan ini muncul masukan  $x=0$ , maka keluaran  $Z=0$ . Karena masukan ini mungkin merupakan awal dari pola yang dicari, kita tandai keadaan-berikut akibat masukan ini sebagai keadaan baru dan kita namakan saja  $S_1$ . Bila pada keadaan  $S_0$  muncul masukan 1, yang bukan awal pola masukan yang dicari, maka masukan ini dapat kita abaikan dan rangkaian tidak berubah keadaan, tetap pada keadaan  $S_0$  menantikan masukan 0 awal pola. Perubahan keadaan yang sudah dibicarakan sampai saat ini dapat digambarkan dalam bentuk diagram keadaan seperti yang ditunjukkan pada Gambar 10.4 (a).

Kalau pada keadaan  $S_1$  muncul masukan  $x=0$ , maka keadaan menerima dua 0 sama saja dengan keadaan menerima satu 0 sehingga keadaan tetap pada keadaan  $S_1$ . Tetapi bila dalam keadaan ini muncul  $x=1$ , maka telah diterima deretan 01 yang merupakan awal pola yang dicari. Karena itu keadaan untuk deretan masukan ini perlu diingat sebagai keadaan baru, kita namakan saja  $S_2$ . Dengan demikian diagram keadaan kita menjadi seperti yang ditunjukkan pada Gambar 10.4 (b).

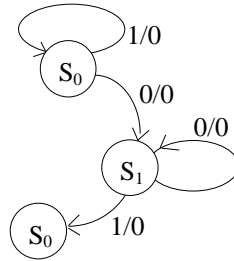
Kalau pada keadaan  $S_2$  muncul masukan  $x=1$ , artinya telah terjadi deretan 011, maka pola yang terjadi bukanlah pola yang dicari. Untuk dapat terbentuk pola yang dicari, maka masukan berikutnya harus mulai lagi dari 0. Ini berarti bahwa keadaan sesudah deretan 011 ini sama saja dengan belum menerima masukan sama sekali, yaitu  $S_0$ . Jadi masukan ini membuat keadaan berubah dari  $S_2$  menjadi  $S_0$ . Tetapi bila pada keadaan  $S_2$  muncul masukan 0, maka masukan ini melengkapi pola yang dicari sehingga keluaran harus  $z=1$ . Pelengkapan pola yang ditutup dengan masukan 0 ini juga dapat merupakan awal pola yang dicari berikutnya, yaitu keadaan telah menerima satu 0, yaitu keadaan  $S_1$ . Jadi, pelengkapan pola ini juga mengubah keadaan dari  $S_2$  ke  $S_1$ . Diagram keadaan yang kita peroleh sekarang menjadi seperti yang ditunjukkan pada Gambar 10.4 (c).

Setelah menyelesaikan Gambar 10.4 (c), kita perlu menilik ulang apakah

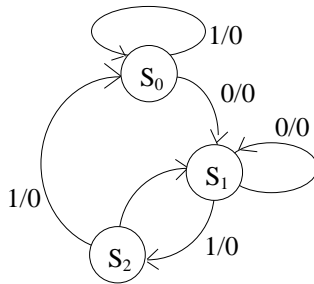
semua kombinasi masukan untuk setiap keadaan telah terwakili dalam diagram keadaan tersebut. Ternyata dalam Gambar 10.4 (c) tersebut semuanya telah terwakili dan diagram tersebut merupakan diagram keadaan yang lengkap rangkaian yang dicari. Dari diagram keadaan ini dapat diperoleh data yang diperlukan untuk menyusun tabel keadaan rangkaian seperti yang ditunjukkan pada Gambar 10.4(d).



(a)



(b)



(c)

Keadaan sekarang	Keadaan berikut		Keluaran sekarang	
	X=0	X=1	X=0	X=1
S <sub>0</sub>	S <sub>1</sub>	S <sub>0</sub>	0	0
10	S <sub>1</sub>	0	S <sub>1</sub>	S <sub>2</sub>
	0	0		
	S <sub>2</sub>		S <sub>1</sub>	S <sub>0</sub>
01	00	1	0	

AB	A <sup>+</sup> B <sup>+</sup>		Z	
	X=0	X=1	X=0	X=1
00	01	00	0	0
0	0	01		01
1	0			10

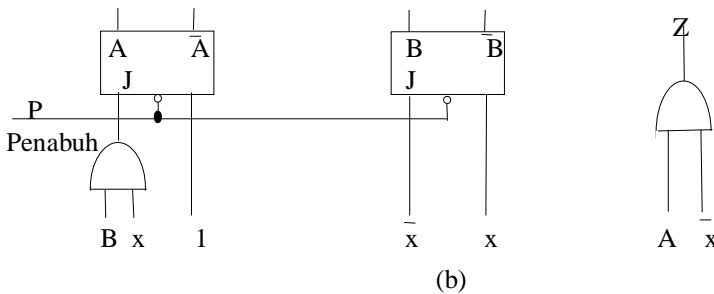
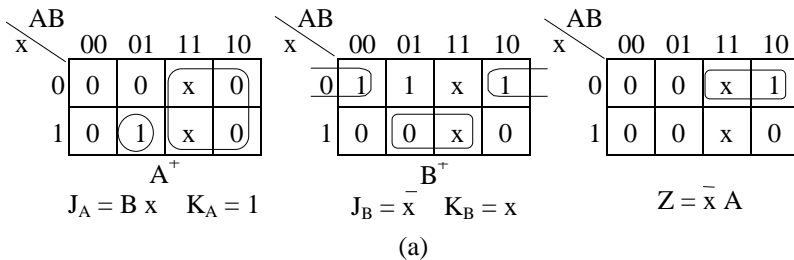


(d)

(e)

Gambar 10.4. Diagram dan Tabel Keadaan detektor urutan.

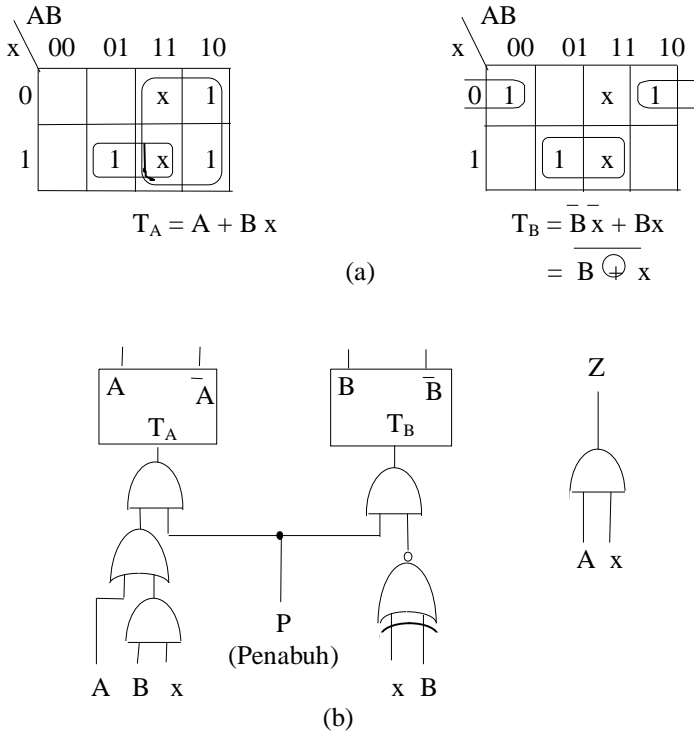
Karena 1 flip-flop mempunyai 2 keadaan dan rangkaian membutuhkan 3 keadaan, maka untuk merealisasikan rangkaian ini dibutuhkan 2 flip-flop, kita sebut saja flip-flop A dan B. Dengan memilih  $A=B=0$  untuk keadaan  $S_0$ ;  $A=0, B=1$  untuk keadaan  $S_1$  dan  $A=1, B=0$  untuk keadaan  $S_2$ , maka tabel keadaan pada Gambar 10.4 (d) dapat diubah ke bentuk tabel keadaan seperti yang ditunjukkan dalam Gambar 10.4 (e). Dari tabel keadaan terakhir ini dengan mudah dapat ditentukan persamaan masukan masing-masing flip-flop dalam rangkaian. Dengan memakai flip-flop JK, maka persamaan masukan diperoleh seperti yang ditunjukkan pada rangkaian Gambar 10.5(a) dan rangkaiannya ditunjukkan pada Gambar 10.5(b).



Gambar 10.5. Peta keadaan dan rangkaian detektor urutan dengan flip-flop JK

Bila direalisasikan dengan flip-flop T, maka dari Gambar 10.4 dapat dibuat peta masukan dan rangkaiannya seperti ditunjukkan pada Gambar 10.6 (a) dan (b). Perhatikan bahwa rangkaian di atas adalah rangkaian Mealy (keluarannya ditentukan oleh masukan bersama-sama dengan keadaan-sekarang). Rangkaian itu dapat juga direalisasikan sebagai rangkaian Moore (keluarannya hanya fungsi

dari keadaan-sekarang). Diagram keadaan mesin Moore hampir sama dengan diagram mesin Mealy, bedanya hanya pada penempat-an variabel keluaran dalam diagram.



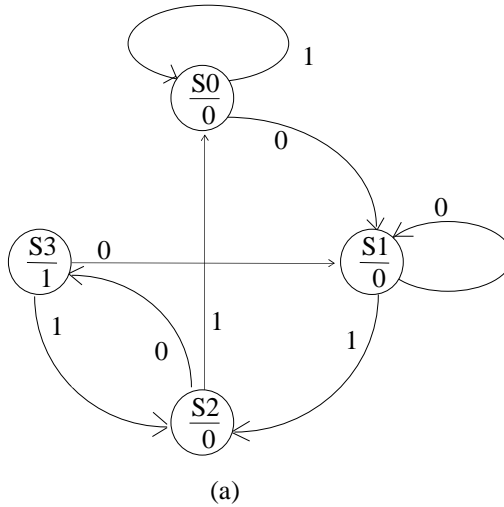
Gambar 10.6. Peta masukan dan rangkaian dengan flip-flop T

Kalau dalam mesin Mealy keluaran ditempatkan di bawah harga masukan maka pada mesin Moore, keluaran ditempatkan di bawah simbol keadaan di dalam lingkaran keadaan. Ini telah dijelaskan dalam bab sebelumnya. Diagram keadaan rangkaian Moore ditunjukkan pada Gambar 10.7.

Kalau dalam mesin Mealy, masukan 0 pada keadaan  $S_2$  akan mengubah keadaan menjadi  $S_1$ , dalam mesin Moore perubahan keadaan dari  $S_2$  ke  $S_1$  ini tidak dimungkinkan karena keluaran pada  $S_1$  adalah 0 sedangkan untuk urutan masukan ini seharusnya  $Z = 1$ . Karena itu dalam mesin Moore ini, masukan 0 pada keadaan  $S_2$  harus di arahkan ke keadaan lain yang mempunyai keluaran  $z =$

1, yaitu  $S_3$ .

Perhatikan bahwa mesin Moore membutuhkan 4 keadaan (bandingkan dengan 3 keadaan dalam mesin Mealy). Walaupun demikian, cacah flip-flop yang dibutuhkan masih sama, yaitu 2. Persamaan masukan flip-flop dan diagram rangkaiannya tentunya sudah dapat diperoleh dengan mudah dari Gambar 10.7(c).



Keadaan sekarang	Keadaan-berikut		Keluaran sekarang (Z)
	x = 0	x = 1	
$S_0$	$S_1$	$S_0$	0
$S_1$	$S_1$	$S_2$	0
$S_2$	$S_3$	$S_0$	0
$S_3$	$S_1$	$S_2$	1

(b)

A B	$A^+ B^+$		Z
	x=0	x=1	
0 0	0 1	0 0	0
0 1	0 1	1 0	0
1 0	1 1	0 0	0
1 1	0 1	0 1	1

(c)

Gambar 10.7. Detektor urutan sebagai mesin Moore.

### 10.3 Penyederhanaan Tabel Keadaan

Kalau minimisasi dalam rangkaian kombinasi berusaha menggabungkan sukumin/sukumax sebanyak-banyaknya untuk memperoleh penyusun dengan

literal paling sedikit, yang berarti cacah masukan yang paling kecil, maka minimisasi dalam rangkaian berurut adalah mencari cacah keadaan yang paling sedikit yang berarti mencari cacah flip-flop yang paling sedikit.

Tabel keadaan dan diagram keadaan dalam 2 macam desain yang telah diuraikan dalam sub-bab terdahulu dapat diperoleh dengan tidak terlalu sulit dan sudah memberikan cacah keadaan minimum. Dalam banyak hal, cacah keadaan yang diperoleh dalam perencanaan tahap awal suatu rangkaian berurut belumlah minimum dan masih dapat dikurangi. Kerena itu, sebelum merealisasikan tabel keadaan suatu rangkaian berurut masih perlu diadakan penilikan apakah cacah keadaan dalam tabel itu sudah minimum. Ini perlu sebab semakin sedikit cacah keadaan yang harus direalisasikan, pada umumnya, semakin sedikit gerbang logika yang dibutuhkan dan ada kalanya kebutuhan flip-flop juga semakin sedikit.

Diagram keadaan untuk rangkaian dengan hanya beberapa keadaan biasanya mudah dibentuk. Tetapi untuk rangkaian dengan keadaan yang banyak, pembentukan diagram keadaan tersebut menjadi jauh lebih sulit, dapat memancing kesalahan dalam menentukan arah/tujuan transisi (peralihan) keadaan. Hal ini akan lebih sulit lagi bila dalam rangkaian itu terdapat lebih dari 1 masukan dan lebih dari 1 keluaran (MIMO, Multiple Input Multiple Output). Dalam hal seperti ini, biasanya kita beranjak dari tabel keadaan. Dalam pembentukan tabel keadaan ini, umumnya lebih mudah bila kita tidak segera mencari keterkaitan antar keadaan secara rinci, tetapi membiarkan setiap kemungkinan masukan menuju keadaan yang baru (yang belum didefinisikan sebelumnya) dan kemudian disederhanakan dengan menghilangkan keadaan yang mubazir (redundant), yaitu keadaan yang dapat digantikan oleh keadaan lain yang setara. Dua cara penyederhanaan yang akan diuraikan dalam bab ini adalah:

- Pencocokan baris (Row Matching)
- Pencocokan dengan peta pasangan (Pair Chart)

### 10.3.1 Pencocokan Baris

Dalam metode ini, setiap baris dalam tabel keadaan dicocokkan dengan baris lain. Dua baris dikatakan sama/ setara (equivalent) bila **keadaan-berikut dan keluaran** pada kedua baris itu sama. Untuk semua baris yang sama digunakan hanya satu sedangkan baris yang lainnya merupakan baris keadaan mubazir dan harus dihilangkan/dihapus, dan setiap kemunculannya sebagai keadaan-berikut pada baris lain digantikan dengan keadaan setaranya. Penyederhanaan selesai bila tidak ada lagi dua baris yang sama.

Misalkan kita hendak merancang detektor urutan yang akan memberikan

keluaran 1 bila deretan masukan "110" atau "101" terjadi. Untuk lebih menjelaskan perilaku rangkaian ini, dapat dibuatkan deretan contoh masukan sebagai berikut:

$$x = 00101110010100110100011100$$

$$z = 0000101000100001100000010$$

Perhatikan bahwa 1 terakhir dalam pola "101" dapat menjadi 1 pertama dalam pola "110" dan 10 penutup dalam pola "110" dapat pula menjadi awal bagi pola "101".

Untuk memulai penyusunan tabel keadaan, kita misalkan keadaan saat sebelum ada masukan yang diterima (keadaan reset) sebagai keadaan awal dan kita sebut keadaan A. Dengan munculnya masukan pertama, rangkaian akan berubah keadaan ke dua kemungkinan keadaan baru: keadaan B bila masukan  $x=0$  dan keadaan C bila  $x=1$ . Masukan ke-2 dapat mengubah keadaan rangkaian ke salah satu dari 4 keadaan baru, sebut saja D, E, F, atau G masing-masing untuk urutan masukan 00, 01, 10, dan 11. Dengan meneruskan urutan-urutan masukan selanjutnya akan kita peroleh tabel keadaan yang ditunjukkan pada Tabel 10.1.

**Tabel 10.1. Tabel keadaan detektor urutan 110 dan 101**

Urutan masukan	Keadaan sekarang	Keadaan-berikut		Keluaran	
		x = 0	x = 1	x=0	x=1
reset	A	B	C	0	0
0	B	D	E	0	0
1	C	F	G	0	0
00	D	D	E	0	0
01	E	F	G	0	0
10	F	D	E	0	1
11	G	F	G	1	0

Karena pola yang dideteksi terdiri atas deretan hanya 3 masukan, yang terlengkapi saat masukan ke-3 muncul dalam keadaan telah menerima 2 masukan, maka kita mendefinisikan keadaan baru cukup untuk mengingat sampai deretan 2 masukan berurut. Perhatikan bahwa setelah munculnya masukan ke 3, keadaan-berikut sudah dapat diwakili oleh keadaan untuk 2 masukan. Walaupun tabel diteruskan dengan mendefinisikan keadaan baru untuk deretan masukan

yang lebih panjang, hasilnya akan sama saja. Masukan 1 pada keadaan telah menerima deretan 10 (keadaan F), misalnya akan membentuk deretan 101. Dalam menunggu masukan berikutnya, keadaan telah menerima deretan 101 ini sama saja dengan keadaan baru menerima deretan 01 (keadaan E) sehingga dalam tabel ditunjukkan bahwa keadaan-berikut setelah menerima masukan 1 pada keadaan F adalah keadaan E.

Tabel keadaan ini tentunya belumlah memberikan cacah keadaan yang minimum. Secara sekilas saja dapat dilihat bahwa keadaan B identik dengan D. Jadi perlu dilakukan penyederhanaan.

Langkah pertama dalam penyederhanaan dengan metoda pencocokan baris adalah menilik keadaan yang potensial (mempunyai potensi) untuk sama. Dua keadaan dikatakan potensial sama bila kedua keadaan itu mempunyai keluaran yang sama untuk semua masukan. Dua keadaan sama bila kedua keadaan itu potensial sama dan mempunyai keadaan-berikut yang sama untuk semua masukan. Jadi hanya yang mempunyai keluaran yang sama saja yang perlu dibandingkan. Biasanya lebih menyenangkan bila kita mulai membandingkan baris terakhir dengan semua baris di atasnya yang potensial sama dan melakukan pengantian nama keadaan bila diperoleh ada dua keadaan yang sama.

Baris keadaan F dengan keluaran berikut adalah  $z = 0$  dan 1, dan G keluaran berikut adalah  $z = 1$  dan 0, tidak potensial dengan keadaan lain yang manapun karena masing-masing mempunyai keluaran yang tidak sama dengan keadaan lain. Karena itu, kedua keadaan ini tak perlu dibandingkan. Keadaan yang potensial setara adalah A, B, C, D, E, dan hanya keadaan-keadaan inilah yang perlu dibandingkan satu dengan lainnya.

Dari tabel awal di Tabel 10.1 di atas segera dapat dilihat bahwa syarat supaya pasangan D dan E sama adalah  $D \equiv F$  dan  $E \equiv G$ . Ini jelas tidak terpenuhi dan karena itu pasangan D dan E tidak sama. Syarat supaya  $C \equiv E$  sama adalah  $F \equiv F$  dan  $G \equiv G$ , yang jelas terpenuhi. Karena itu pasangan E dan C setara, baris E dapat dicoret dari tabel serta semua keadaan E sebagai keadaan-berikut di baris lain dapat diganti dengan C seperti yang ditunjukkan pada Tabel 10.2(a).

**Tabel 10.2(a). Tabel 10.1 dengan kesetaraan  $C \equiv E$**

Urutan masukan	Keadaan sekarang	Keadaan-berikut		Keluaran	
		$x = 0$	$x = 1$	$x=0$	$x=1$
reset	A	B	C	0	0
0	B	D	<del>E</del> C	0	0
1	C	F	G	0	0
00	D	D	<del>E</del> C	0	0
.....					

01	E	F	G	0	0	$E \equiv C$
10	F	D	E	0	1	
11	G	F	G	1	0	

Dari Tabel 10.2(a) segera dapat dilihat bahwa  $D \equiv B$  dan baris D dapat dicoret serta semua keadaan D di baris lain dalam tabel diganti dengan B dan tabel keadaan menjadi seperti yang ditunjukkan pada Tabel 10.2 (b).

Dari tabel ini segera lagi terlihat bahwa  $B \equiv A$  dan baris B dapat dicoret, dan diperoleh Tabel 10.2(c). Kesetaraan  $B \equiv A$  ini merupakan kesetaraan terakhir yang dapat diperoleh sehingga Tabel 10.2(c) telah merupakan tabel keadaan minimum. Ternyata dari 7 keadaan semula, tinggal 4 keadaan yang harus dipertahankan, yaitu A, C, F dan G seperti yang ditunjukkan dalam Tabel 10.3.

Kalau tabel keadaan dalam Tabel 10.3 digambarkan diagramnya kita akan mendapat diagram keadaan seperti pada Gambar 10.8. Karena 4 keadaan tepat dapat dinyatakan oleh 2 flip-flop, maka ke-4 keadaan ini dapat dinyatakan dengan biner 0-3 misalnya  $A = 00$ ,  $C = 01$ ,  $F = 10$  dan  $G = 11$ . Dengan penetapan keadaan ini, maka persamaan masukan flip-flop sudah dapat diturunkan dari tabel keadaan ini.

**Tabel 10.2(b). Tabel 10.1 dengan kesetaraan  $C \equiv E$  dan  $D \equiv B$**

Urutan masukan	Keadaan sekarang	Keadaan-berikut		Keluaran		
		x = 0	x = 1	x=0	x=1	
reset	A	B	C	0	0	
0	B	<del>B</del>	<del>C</del>	0	0	
1	C	F	G	0	0	
00	D	D	<del>C</del>	0	0	$D \equiv B$
01	E	F	G	0	0	$E \equiv C$
10	F	D	E	0	1	
11	G	F	G	1	0	

**Tabel 10.2(c). Tabel 10.1 dengan kesetaraan  $C \equiv E$ ,  $D \equiv B$  dan  $B \equiv A$**

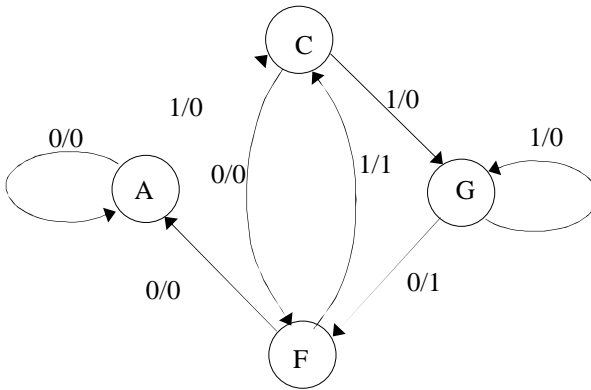
Urutan masukan	Keadaan sekarang	Keadaan-berikut		Keluaran	
		x = 0	x = 1	x=0	x=1



		10.3.2 Peta Pasangan				
reset	A	B	C	0	0	
0	B	<del>D</del> B	<del>E</del> C	0	0	B ≡ A
1	C	F	G	0	0	
00	D	D	<del>E</del> C	0	0	D ≡ B
01	E	F	G	0	0	E ≡ C
10	F	D	E	0	1	
11	G	F	G	1	0	

**Tabel 10.3. Tabel 10.1 yang disederhanakan.**

Urutan masukan	Keadaan sekarang	Keadaan-berikut		Keluaran	
		x = 0	x = 1	x=0	x=1
reset	A	A	C	0	0
1	C	F	G	0	0
10	F	A	C	0	1
	11	G	F	G	1



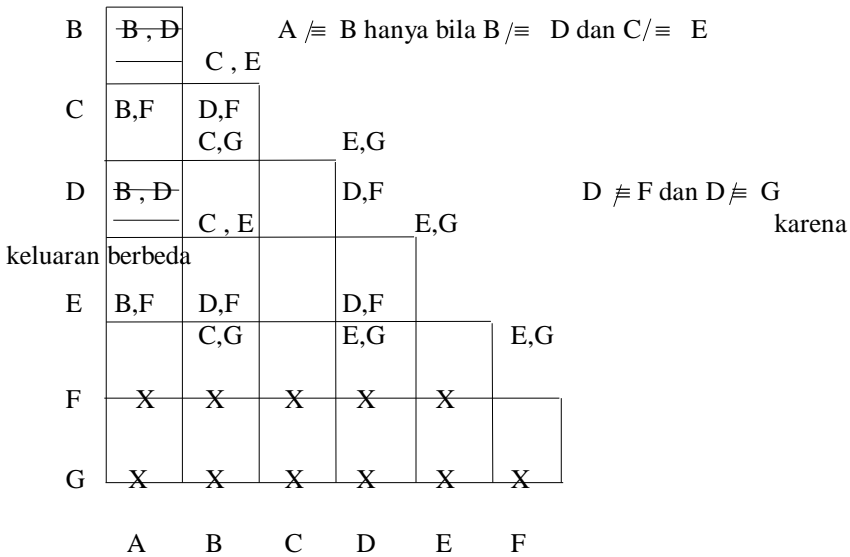
Gambar 10.8. Diagram keadaan detektor urutan 110 dan 101

**10.3.2 Peta Pasangan**

Penentuan setara dengan cara pencocokan baris (row matching) sering tidak cukup untuk memperoleh keadaan minimum. Cara yang lebih baik dalam penentuan kesetaraan keadaan adalah dengan memakai tabel penyusunan (implication table) yang disebut juga peta pasangan (pair chart). Peta pasangan merupakan susunan kotak-kotak yang menghubungkan satu keadaan dengan keadaan lain secara berpasangan. Setiap kotak ditandai dengan nama baris dan nama kolom yang merupakan nama-nama keadaan yang dihubungkan kotak bersangkutan dan masing-masing kotak diisi dengan syarat yang harus dipenuhi supaya dua keadaan yang dihubungkannya setara. Bila dua keadaan sudah sama, kotak yang menghubungkannya dikosongkan saja. Bila syarat didalam kotak terpenuhi maka kedua keadaan setara. Setiap kotak yang syarat di dalamnya tak terpenuhi menunjukkan kedua keadaan yang dihubungkannya

tidak setara dan dicoret atau diisi dengan tanda X. Di akhir pencocokan, semua kotak yang tidak dicoret menunjukkan kesetaraan sehingga keadaan minimum dapat diperoleh dengan mengambil hanya satu dari setiap pasangan yang setara.

Untuk lebih jelasnya, kita perhatikan contoh yang digunakan dalam penyederhanaan dengan pencocokan baris sebelumnya. Peta pasangan untuk tabel keadaan Tabel 10.1 dapat dibuat sebagai ditunjukkan pada Gambar 10.9.



Gambar 10.9. Peta pasangan detektor urutan 110 dan 101

Dalam peta tersebut, setiap kotak yang mewakili keadaan yang tak potensial sama segera diberi tanda silang (tanda X), seperti yang menghubungkan A dengan F dan A dengan G (keluaran berbeda). Kotak-kotak yang menghubungkan keadaan yang potensial sama diisi dengan syarat yang harus dipenuhi untuk setara. Misalnya, supaya A setara dengan B ( $A \equiv B$ ) harus dipenuhi  $B \equiv D$  dan  $C \equiv E$  dan kedua syarat ini diisikan ke dalam kotak kolom A baris B. Kotak lain dapat diisi dengan mengacu Tabel 10.1. Perhatikan bahwa kotak-kotak (B,D) dan (C,E) tidak diisi apa-apa karena syarat kesetaraan mereka telah terpenuhi, yaitu syarat  $D \equiv D$  dan  $E \equiv E$  untuk kesetaraan pasangan (B,D) dan syarat  $F \equiv F$  dan  $G \equiv G$  untuk kesetaraan pasangan (C,E). Karena itu, semua persyaratan kesetaraan (B,D) dan (C,E) dapat dicoret dari kotak lain, misalnya di kotak (A,D) dan di kotak (A,B).

Pengamatan atas peta dapat dilihat bahwa A dan C tidak akan setara karena  $B \neq F$  dan  $C \neq G$ , [kotak (B,F) dan (C,G) telah berisi tanda X). Karena itu kotak (A,C) sudah dapat diisi tanda X. Begitu juga kotak-kotak (A,E), (B,C), (B,E), (C,D), (D,E) sudah dapat diisi X. Dengan memberi tanda X ke dalam kotak-kotak yang tidak setara ini, maka kita akan memperoleh peta pasangan baru seperti pada Gambar 10.10. Perhatikan bahwa syarat (B,D) dan (C,E) tidak muncul lagi dalam peta ini. Semua kotak yang tidak diberi tanda X pada peta terakhir ini menunjukkan kesetaraan pasangan keadaan yang bersesuaian dengan kotak itu.

B						
C	X	X				
D			X			
E	X	X		X		
F	X	X	X	X	X	
G	X	X	X	X	X	X
	A	B	C	D	E	F

Gambar 10.10. Penyederhanaan peta pasangan Gambar 10.9

Dari peta terakhir ini dapat disimpulkan bahwa kesetaraan yang terdapat dalam tabel keadaan sebelumnya adalah:

$$A \equiv D, A \equiv B, B \equiv D \text{ atau } D \equiv B \equiv A$$

dan

$$C \equiv E$$

Bila kesamaan ini digunakan untuk menggantikan D dan B dengan A, dan E dengan C, maka Tabel 10.1 sebelumnya akan menjadi Tabel 10.4 yang kembali menjadi Tabel 10.3 bila baris D, B, dan E dihilangkan.

Langkah-langkah penentuan keadaan setara dengan metoda tabel penyusunan/peta pasangan yang diterangkan di atas dapat diringkas sebagai berikut ini:

1. Bentuk peta pasangan yang terdiri atas kotak-kotak dengan sisi yang merupakan nama keadaan, satu kotak untuk setiap pasangan keadaan. Bagian di atas diagonal tidak disertakan.
2. Bandingkan setiap pasangan dalam tabel keadaan. Bila keluaran untuk keadaan  $i$  dan keadaan  $j$  berbeda, maka ke dalam kotak  $(i,j)$  diisikan tanda X yang menyatakan bahwa  $i \equiv j$ . Bila keluaran tersebut sama, maka ke dalam kotak  $(i,j)$  diisikan syarat kesetaraan pasangan yang harus dipenuhi (implied pair), yaitu  $m = n$ , ditulis " $m,n$ ", dengan  $m$  dan  $n$  masing-masing merupakan keadaan-berikut daripada keadaan  $i$  dan  $j$  untuk masukan yang sama.

**Tabel 10.4. Tabel keadaan 10.1 yang disederhanakan**

Urutan masukan	Keadaan sekarang	Keadaan-berikut		Keluaran		
		x = 0	x = 1	x=0	x=1	
reset	A	A	C	0	0	
0	B	A	C	0	0	← buang
1	C	F	G	0	0	
00	D	A	C	0	0	← buang
01	E	F	G	0	0	← buang
10	F	A	C	0	1	
11	G	F	G	1	0	

3. Telitilah kotak demi kotak dalam peta. Bila kotak  $(i,j)$  mengandung pasangan  $(m,n)$  sedangkan kotak  $(m,n)$  mengandung X maka berikan tanda X pada kotak  $(i,j)$ . Bila kotak  $(m,n)$  kosong, yang berarti keadaan  $m$  setara dengan keadaan  $n$ , maka semua syarat kesetaraan " $m,n$ " dalam kotak lain dicoret (bukan disilang/diberi X) sebagai tanda bahwa kesetaraan pasangan tersebut tidak lagi merupakan syarat karena telah dipenuhi.
4. Bila pada langkah ke-3 masih ada tanda X yang ditambahkan, ulangi langkah ke-3 tersebut sampai tidak ada lagi penambahan X.
5. Setiap kotak  $(i,j)$  yang tidak mengandung X menunjukkan kesetaraan  $i \equiv j$ .

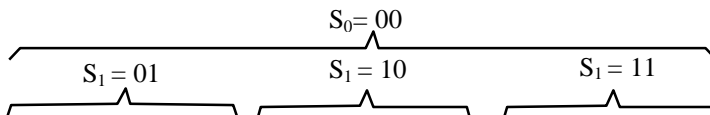
### 10.4 Penetapan Keadaan

Dalam uraian sebelumnya, penyederhanaan rangkaian berurut memusatkan perhatian hanya pada pencarian cacah keadaan sekecil mungkin, tanpa

memperhitungkan harga rangkaian masukan setiap flip-flop. Pengurangan cacah keadaan rangkaian berurut sering harus dibayar dengan rangkaian kombinasi yang malah lebih mahal dari pada penghematan yang diperoleh dari pengurangan cacah keadaan. Di samping itu, penggerbangan yang dibutuhkan untuk satu himpunan keadaan biner *sering sekali* jauh lebih rumit dan mahal daripada yang dibutuhkan untuk himpunan keadaan biner yang lain walaupun untuk himpunan keadaan hasil penyederhanaan yang sama. Penentuan harga biner untuk setiap keadaan akhir yang akan menghasilkan penggerbangan paling minimal disebut **“Penetapan keadaan” (State Assignment)**. Berbagai metoda penetapan keadaan sudah ditawarkan, tetapi sampai sejauh ini belum ada metoda yang dapat memberikan penentuan keadaan yang selalu optimal untuk semua rangkaian. Metoda coba-coba (trial and error) merupakan salah satu metoda yang berlaku umum. Tetapi tentunya akan membutuhkan perhitungan yang sangat melelahkan bila cacah keadaannya sangat besar. Metoda yang diajukan oleh James R. **Story**, Harold J. **Harrison**, dan Erwin A. **Reinhard** (1972), yang disebut metoda **SHR** (singkatan dari nama ketiga penemunya: Story, Harrison, Reinhard), juga merupakan metoda coba-coba yang selalu memberikan hasil optimal dengan langkah coba-coba (trial) yang sangat dikurangi. Tetapi metoda SHR berlaku hanya untuk rangkaian berurut yang menggunakan flip-flop JK.

Dengan satu flip-flop, dapat disajikan 2 keadaan, yaitu keadaan: 0 dan 1. Dengan 2 flip-flop, ada 4 keadaan yang dapat disajikan, yaitu: 00, 01, 10 dan 11. Dengan 3 flip-flop, ada 8 keadaan yang dapat disajikan, dan seterusnya. Secara singkat, dengan  $n$  flip-flop dapat disajikan  $2^n$  keadaan. Jika keseluruhan keadaan yang mungkin digunakan untuk menyajikan keadaan rangkaian, maka tidak ada masalah. Sebaliknya, jika dari semua keadaan yang mungkin hanya sebagian yang dimanfaatkan untuk menyajikan keadaan rangkaian, maka terdapat banyak kombinasi keadaan yang mungkin. Untuk rangkaian dengan 3 keadaan, misalnya  $S_0$ ,  $S_1$ , dan  $S_2$ , dibutuhkan 2 flip-flop untuk merealisasikannya, sebut saja flip-flop A dan B. Tetapi 2 flip-flop menyediakan 4 keadaan, sehingga cukup banyak kemungkinan kombinasi pasangan keadaan yang dapat dipilih untuk rangkaian tersebut. Dengan memilih  $AB=00$  untuk keadaan  $S_0$ , maka keadaan  $S_1$  dan  $S_2$ , dapat dipilih dari 3 keadaan yang lain. Bila salah satu dari 3 keadaan tersebut ditentukan untuk keadaan  $S_2$ , misalnya 01, maka untuk  $S_3$  tinggal 2 keadaan yang dapat dipilih, yaitu 10 dan 11. Jadi, untuk  $S_0=00$  terdapat 6 kombinasi keadaan untuk  $(S_0, S_1, S_2)$ :

(00,01,10); (00,01,11); (00,10,01); (00,10,11);(00,11,01); (00,11,10)



(00,01,10); (00,01,11); (00,10, 01); (00,10,11); ( 00,11,01); (00,11,10)

Sejumlah keadaan yang sama dapat lagi dibentuk untuk masing-masing  $S_0=01$ ,  $S_0= 10$  dan  $S_0= 11$ . Dalam Tabel 10.5 ditunjukkan 24 kombinasi yang mungkin. Tetapi jika kita perhatikan, beberapa kombinasi akan memberikan harga realisasi yang sama, misalnya, harga realisasi kombinasi keadaan (00, 01, 10) sama dengan kombinasi keadaan (00, 10, 01), harga (00, 01, 11) sama dengan (00, 10, 11), dan seterusnya.

Tabel 10.5. Kombinasi 3 keadaan untuk 2 flip-flop

Keadaan Rangkaian	Keadaan Flip-flop																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$S_0$	00	00	00	00	00	00	01	01	01	01	01	01	10	10	10	10	10	10
$S_1$	01	01	10	10	11	11	00	00	10	10	11	11	00	00	01	01	11	11
$S_2$	10	11	01	11	01	10	10	11	00	11	00	10	01	11	00	11	00	01

Keadaan Rangkaian	19	20	21	22	23	24
$S_0$	11	11	11	11	11	11
$S_1$	00	00	01	01	10	10
$S_2$	01	10	00	10	00	01

Pada umumnya, *pertukaran kolom dalam setiap pemilihan kombinasi keadaan tidak mengubah harga realisasi* (dalam Tabel 10.5 ada 2 kolom, A dan B, untuk setiap pilihan keadaan). Lebih lanjut, *untuk flip-flop simetris RS, JK, dan T, mengkomplemenkan satu atau lebih kolom tidak akan mengubah harga realisasi*. Jadi, harga realisasi (00, 01, 10) sama dengan (10, 11, 01), yaitu mengkomplemenkan kolom A; sama dengan (01, 00, 11), yaitu mengkomplemenkan kolom B; sama dengan (11, 10, 01), yaitu mengkomplemenkan kolom A dan kolom B. Untuk flip-flop D, pengkomplemenan boleh jadi akan membutuhkan tambahan gerbang inverter (NOT). Dengan memanfaatkan sifat-sifat ini, maka cacah coba-coba untuk mencari kombinasi keadaan yang minimal akan sangat dikurangi. Untuk penetapan keadaan dalam Tabel 10.5 di atas, dapat diperoleh kesamaan keadaan berikut ini:

$$\begin{aligned}
 1 &= 3 = 8 = 11 = 14 = 17 = 22 = 24 \\
 2 &= 4 = 7 = 12 = 13 = 18 = 21 = 23
 \end{aligned}
 \tag{10.1}$$

$$5 = 6 = 9 = 10 = 15 = 16 = 19 = 20$$

Ini berarti bahwa dari ke-24 kemungkinan kombinasi keadaan yang dapat dipilih, kita cukup mencoba dan memilih yang termurah dari 3 kombinasi saja, yaitu kombinasi 1, 2, dan 5.

Untuk melihat pengaruh penetapan keadaan ini, kita perhatikan suatu rangkaian berurut dalam Gambar 10.4 di depan yang ditunjukkan lagi berikut ini.

Keadaan sekarang	Keadaan berikut		Keluaran sekarang	
	X=0	X=1	X=0	X=1
S <sub>0</sub>	S <sub>1</sub>	S <sub>0</sub>	0	0
S <sub>1</sub>	S <sub>1</sub>	S <sub>2</sub>	0	0
S <sub>2</sub>	S <sub>1</sub>	S <sub>0</sub>	1	0

Karena adanya 3 keadaan dibutuhkan 2 flip-flop. Tetapi 2 flip-flop juga dapat menyatakan 4 keadaan tanpa menambah gerbang. Pemilihan harga biner keadaan yang ditunjukkan pada Gambar 10.4(b) adalah pemilihan langsung secara berurutan. Sebenarnya terdapat beberapa kemungkinan lain untuk penetapan keadaan ini. Dengan menggunakan temuan seperti pada persamaan (10.1) di atas, kita cukup mencoba 3 pilihan keadaan: 1, 2 dan 5 dengan tabel keadaan seperti Gambar 10.11.

AB	A <sup>+</sup> B <sup>+</sup>		Z	
	X=0	X=1	X=0	X=1
00	01	00	0	0
01	01	10	0	0
10	01	00	1	0

(a)

AB	A <sup>+</sup> B <sup>+</sup>		Z	
	X=0	X=1	X=0	X=1
00	01	00	0	0
01	01	11	0	0
11	01	00	1	0

(b)

AB	A <sup>+</sup> B <sup>+</sup>		Z
	X=0	X=1	X=0 Y=1



00	11	00	0	0
11	11	10	0	0
10	11	00	1	0

(c)

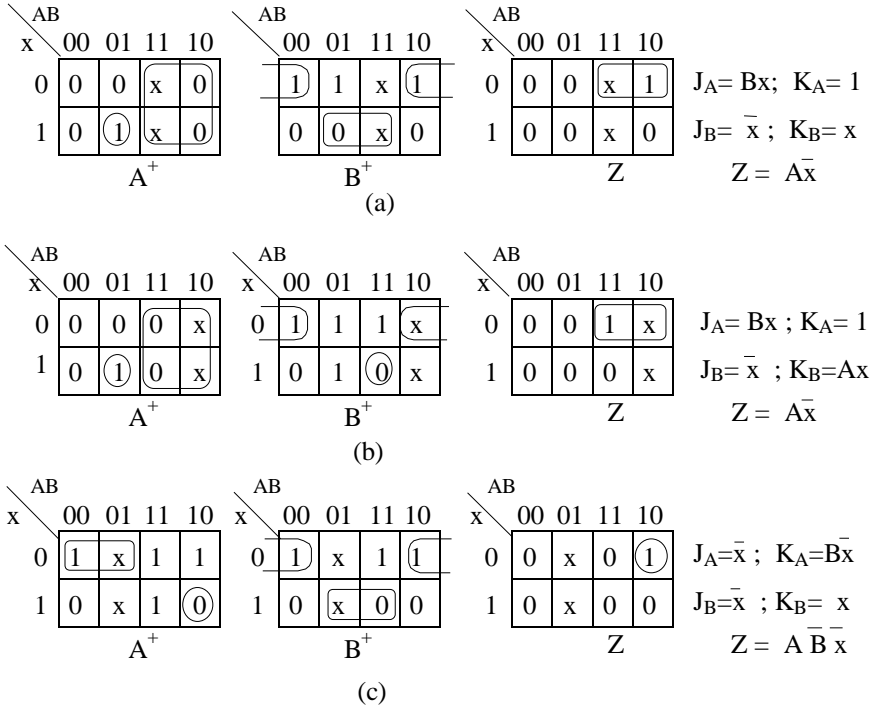
**Gambar 10.11.** Tabel Keadaan-berikut

(a)  $(S_0, S_1, S_2) = (00, 01, 10)$

(b)  $(S_0, S_1, S_2) = (00, 01, 11)$

(c)  $(S_0, S_1, S_2) = (00, 11, 01)$

Masing-masing pilihan ini akan memberikan persamaan masukan seperti yang ditunjukkan pada Gambar 10.12. Dari sini dapat dilihat bahwa pilihan 1 merupakan realisasi yang paling murah, membutuhkan 2 gerbang AND 2-masukan dan 1 gerbang NOT, sedangkan pilihan 2 membutuhkan 3 AND 2-masukan dan 1 NOT, dan pilihan 3 membutuhkan 1 AND 2-masukan, 1 AND 3-masukan dan 2 gerbang NOT.



Gambar 10.12. Pers. masukan dan keluaran flip-flop JK untuk Gambar 10.11  
 (a)  $(S_0, S_1, S_2) = (00, 01, 10)$ ; (b)  $(S_0, S_1, S_2) = (00, 01, 11)$   
 (c)  $(S_0, S_1, S_2) = (00, 11, 01)$

Perlu dicatat bahwa pertambahan keadaan dan pertambahan variabel keadaan (direalisasikan dengan 1 flip-flop) akan memberikan pertambahan yang jauh lebih besar dalam pilihan kombinasi keadaan yang mungkin. Sebagai contoh, untuk 3 peubahkeadaan A, B, dan C, dapat memberikan 8 keadaan  $S_0$  s/d  $S_7$ . Bila dari ke-8 keadaan ini digunakan hanya 5 keadaan, maka terdapat

140 macam kombinasi yang mungkin dibentuk dengan 15 pilihan keadaan yang berbeda; bila yang digunakan 6 keadaan, terdapat 420 macam kombinasi yang mungkin dibentuk dengan 25 pilihan keadaan yang berbeda; dan bila yang digunakan 7 atau 8 keadaan, terdapat 840 macam kombinasi yang mungkin dibentuk dengan 35 pilihan keadaan yang berbeda. Pertambahan kombinasi keadaan yang berbeda itu akan semakin tajam dengan penambahan keadaan yang digunakan dan peubah (variabel) keadaan. Tentunya tidak praktis mencoba semua kombinasi yang mungkin walaupun hanya yang berbeda. Untuk membantu penetapan keadaan, dapat digunakan pedoman berdasarkan keberdekatan berikut ini:

1. Keadaan-keadaan yang untuk satu masukan mempunyai keadaan-berikut yang sama hendaknya diberikan keadaan yang berdekatan (adjacent).
2. Keadaan-keadaan yang merupakan keadaan-berikut bagi keadaan yang sama hendaknya diberikan keadaan yang berdekatan.
3. Keadaan-keadaan yang mempunyai keluaran yang sama untuk suatu masukan hendaknya diberikan keadaan yang berdekatan. Pedoman ini digunakan dalam penyederhanaan fungsi keluaran.

Keberdekatan yang dimaksud dalam pedoman di atas adalah keberdekatan seperti pada pemetaan Karnaugh, yaitu keadaan yang berbeda hanya satu variabel. Penentuan keadaan mengikuti pedoman di atas dimulai dengan mencatat semua pasangan-pasangan keadaan yang seharusnya ditentukan berdekatan menurut pedoman. Keadaan-keadaan itu kemudian digambarkan dalam peta Karnaugh dengan mencoba memenuhi sebanyak mungkin keberdekatan yang dituntut. Penempatan keadaan-keadaan ke dalam peta Karnaugh perlu diperhatikan hal-hal berikut ini:

- Mulailah menempatkan keadaan awal di kotak nol.
- Dahulukanlah memenuhi keberdekatan pedoman 1 dan keberdekatan yang lebih banyak dituntut
- Tempatkanlah 3 atau 4 keadaan yang dituntut berdekatan oleh pedoman pada 4 kotak yang berdekatan.
- Gunakanlah pedoman 3 dalam penyederhanaan peta keluaran, tetapi masih harus mendahulukan pedoman 1 dan 2.

Perhatikan, sebagai contoh, tabel keadaan mesin Mealy yang ditunjukkan dalam Tabel 10.6. Dalam soal ini, ada 7 keadaan sehingga kita membutuhkan 3 peubah keadaan, kita sebut saja  $p, q$ , dan  $r$ , dan keluaran kita sebut  $z$ . Karena dari kemungkinan 8 keadaan yang dapat dinyatakan dengan 3 peubah keadaan hanya 7 keadaan yang dibutuhkan/digunakan, sebenarnya terdapat 35 kombinasi

berbeda (lihat uraian sebelumnya). Ini tentu sangat melelahkan dan untuk menghindarinya kita gunakan pedoman keberdekatan.

Tabel 10.6. Tabel Keadaan Contoh

Keadaan sekarang	Keadaan berikut		Keluaran sekarang	
	X=0	X=1	X=0	X=1
A	B	C	0	0
B	D	C	0	0
C	B	E	0	0
D	F	C	0	0
E	B	G	0	0
F	F	C	1	0
G	B	G	0	1

Keberdekatan:

1. (A,C,E,G), (A,B,D,F), (D,F), (E,G)
2. (B,C), (C,D), (B,E), (C,F)<sub>2X</sub>, (B,G)<sub>2X</sub>

Tuntutan keberdekatan yang disarankan pedoman 1 dan 2 di atas juga ditunjukkan dalam tabel tersebut. Keberdekatan (A,C,E,G) karena sama-sama mempunyai keadaan-berikut B untuk masukan X= 0, keberdekatan (E,G) karena sama-sama mempunyai keadaan-berikut G untuk masukan X= 1, sedangkan keberdekatan (B,C) karena sama-sama merupakan keadaan-berikut bagi A, keberdekatan (C,D) karena sama-sama merupakan keadaan-berikut bagi B, dan seterusnya.

Untuk keadaan awal, A dapat segera kita pilih A = 000 untuk pqr. Beberapa kemungkinan pilihan keadaan yang memenuhi semua keberdekatan ini dapat dicari dengan menggunakan peta Karnaugh. Perhatikan bahwa keberdekatan (A,C,E,G) dan (A,B,D,F) dari pedoman 1 tak dapat dipenuhi secara bersama-sama. Pada Gambar 10.13 ditunjukkan 3 peta yang memenuhi paling banyak keberdekatan pedoman 1 dan 2, tetapi tak dapat menempatkan (B,C), (C,D), dan (B,E) secara keberdekatan.

	pq			
r	00	01	11	10
0	A	C	E	G
1		F	D	B

(a)

- (A,C,E,G), (D,F), (E,G)  
(C,F)<sub>2X</sub>, (B,G)<sub>2X</sub>

	pq			
r	00	01	11	10
0	A	E		D
1	C	G	B	F

(b)

- (A,C,E,G), (D,F), (E,G)  
(C,F)<sub>2X</sub>, (B,G)<sub>2X</sub>

	pq			
r	00	01	11	10
0	A	B	D	F
1		G	E	C

(c)

- (A,B,D,F), (D,F), (E,G)  
(C,F)<sub>2X</sub>, (B,G)<sub>2X</sub>

Gambar 10.13. Peta penetapan keadaan

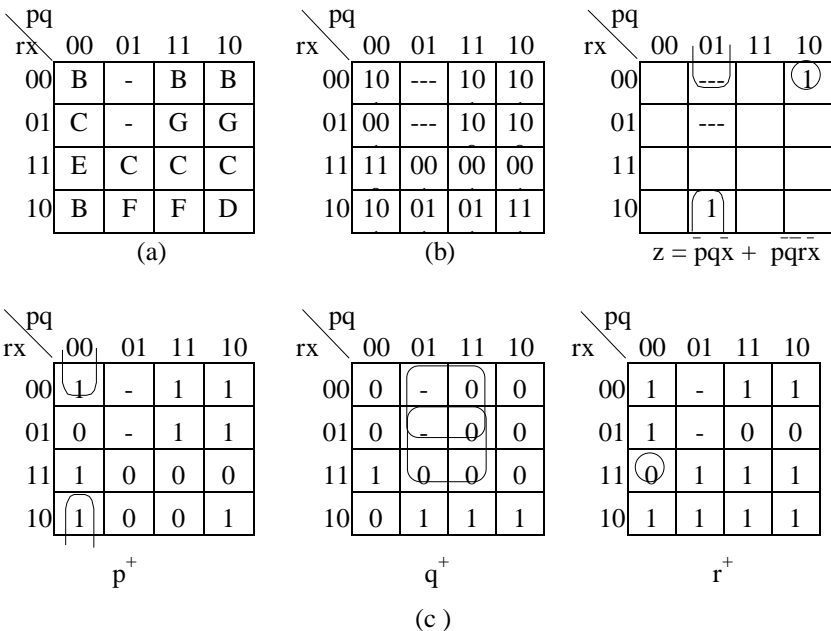
Gambar 10.13 memberikan pilihan keadaan:

Peta (a) : A= 000, B= 101, C= 001, D= 111, E= 110, F= 011, G= 100

Peta (b) : A= 000, B= 111, C= 001, D= 100, E= 010, F= 101, G= 011

Peta (c) : A= 000, B= 010, C= 101, D= 110, E= 111, F= 100, G= 011

Peta keadaan-berikut masing-masing peta penetapan keadaan ini, mengikuti tabel keadaan pada Tabel 10.16, dapat digambarkan seperti pada Gambar 10.14.



Gambar 10.14. Peta keadaan berikut pilihan Gambar 10.13(a).

- (a) Peta dengan nama keadaan
- (b) Peta dengan biner peubah keadaan
- (c) Peta masing-masing peubah

Dari Gambar 10.14 (c) sudah dapat ditentukan persamaan masukan masing-masing flip-flop. Untuk flip-flop JK, dapat diperoleh:

$$\begin{aligned} J_p &= \bar{q} \bar{x} + \bar{q} x & J_q &= \bar{p} r x & J_r &= \bar{p} + \bar{x} \\ K_p &= q r + r x & K_q &= \bar{r} + x & K_r &= \bar{p} \bar{q} x & z &= \bar{p} q \bar{x} + \bar{p} \bar{q} r x \end{aligned}$$

Untuk penetapan keadaan yang lain dapat dilakukan dengan cara yang sama dan dipilih harga rangkaian yang paling murah. Untuk pilihan (a) pada Gambar 10.12 dengan flip-flop JK di atas dapat dilihat bahwa harga realisasinya adalah 4 AND 2-masukan, 3 AND 3-masukan, 1 AND 4-masukan, 5 OR 2-masukan, atau total 13 gerbang 31-masukan.

### 10.5 Soal Latihan

1. Rencanakanlah suatu detektor pasangan (parity) yang akan memberikan keluaran 1 bila cacah masukan 1 berharga ganjil dan keluaran 0 bila cacah masukan 1 berharga genap. Pendeteksian baru dilakukan bila rangkaian telah menerima dua masukan 0. Gunakan flip-flop RS.
2. Sederhanakanlah tabel keadaan-berikut dalam Tabel S10.2 berikut ini baik secara pencocokan baris maupun peta pasangan.

Tabel S10.2

keadaan sekarang	keadaan-berikut		Keluaran sekarang	
	X = 0	X = 1	X = 0	X = 1
a	c	f	0	0
b	d	e	-	0
c	h	g	0	0
d	b	g	0	0
e	e	b	0	1
f	f	a	0	1
g	c	g	0	1
h	c	f	0	0

3. Suatu rangkaian dengan satu masukan dan satu keluaran mempunyai tabel keadaan seperti ditunjukkan Tabel S10.3 berikut ini. Dengan mengambil  $S_1=000$  tentukan keadaan-keadaan yang lain dan tentukan persamaan masukan untuk masing-masing flip-flop dengan menggunakan flip-flop D.

Tabel S10.3

keadaan sekarang	keadaam berikut		Keluaran Z
	X = 0	X = 1	
S <sub>1</sub>	S <sub>5</sub>	S <sub>4</sub>	0
S <sub>2</sub>	S <sub>1</sub>	S <sub>6</sub>	1
S <sub>3</sub>	S <sub>7</sub>	S <sub>8</sub>	1
S <sub>4</sub>	S <sub>7</sub>	S <sub>1</sub>	0
S <sub>5</sub>	S <sub>2</sub>	S <sub>3</sub>	1
S <sub>6</sub>	S <sub>4</sub>	S <sub>2</sub>	0
S <sub>7</sub>	S <sub>6</sub>	S <sub>8</sub>	0
S <sub>8</sub>	S <sub>5</sub>	S <sub>3</sub>	1

4. Sederhanakanlah tabel keadaan-berikut dalam Tabel S10.4 berikut ini dengan menggunakan tabel pasangan. Realisasikanlah rangkaiannya dengan menggunakan:
- flip-flop D
  - flip-flop JK

Tabel S10.4

Keadaan sekarang	Keadaan-berikut		Keluaran Z
	X = 0	X = 1	
A	A	B	1
B	C	E	0
C	F	G	1
D	C	A	0
E	I	G	1
F	H	I	1

G	C	F	0
H	F	B	1
I	C	E	0

5. a. Untuk kedua kemungkinan pilihan keadaan (b) dan (c) dalam Gambar 10.13, tentukanlah harga rangkaiannya jika menggunakan flip-flop JK.
- b. Kerjakan ketiga pilihan dalam gambar tersebut jika menggunakan flip-flop D.
- c. Cobalah rancang dengan memenuhi keberdekatan (B,C), (C,D), dan (B,E) dan salah satu dari pedoman 1.